

## **REMARKS**

The Examiner is thanked for the performance of a thorough search. By this response, Claims 1, 8–10, 13–15, 22–24, and 27–28 are amended. No claims are added. Claims 30 and 32 are canceled. Hence, Claims 1–29 and 31 are pending in this application.

The amendments to the claims do not add any new matter to this application and are supported by the Specification as originally filed. All issues raised in the Office Action mailed November 25, 2009 are addressed hereinafter.

### **I. CLAIM REJECTIONS BASED ON 35 U.S.C. § 103**

Claims 1–32 were rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over U.S. Patent No. 7,099,027 (hereinafter “*Barry*”) in view of U.S. Patent No. 7,202,972 (hereinafter “*Schwier*”). Applicants traverse the rejection. Reconsideration is respectfully requested.

#### **THE *BARRY* REFERENCE**

*Barry*, to the extent relied upon in the Office Action, describes a networked-based distributed print job system in which a print job is routed from an application at one computer system, over a network, and to a distribution node. *Barry* at abstract. The print job is sent to the distribution node in the form of a PDL input. *Barry* at FIG. 8. While at the distribution node, the print job may be merged with additional PDL information, such as a background graphic. *Barry* at FIG. 8; col. 13, lines 10–40.

#### **THE *SCHWIER* REFERENCE**

*Schwier*, to the extent relied upon in the Office Action, describes the conversion of a document from an application format such as Word to a RAW data stream, which may then be sent to a printer device. *Schwier* at FIG. 8. *Schwier* further describes techniques for filtering a merged EMF document into two separate streams, which are then converted to PCL and sent separately to a printer. *Schwier* at FIG. 2.

## CLAIM 1

Claim 1 presently recites that a merge utility receives a request to merge a first merge document in a merge format with a second document in an original format. In response to the request, the merge utility causes the second document to be converted to the merge format, thereby generating a second merge document. The first and second merge documents are then merged to form a composite merge document. The composite merge document is also in the merge format, meaning it can be properly interpreted by an output device without requiring further conversion. The composite merge document is then delivered to the output device.

For example, the merge utility may receive a request to merge a document (an example second document) saved in Word (an example original format) with a background template (an example first merge document) stored as a PostScript file (an example merge format). In response to the request, the merge utility may cause the Word document to be converted to PostScript format, thereby yielding an example second merge document. The merge utility may then merge the converted Word document and the background template into a single PostScript file (the composite merge document). The merge utility may then cause the merged document to be delivered to a printer (an example output device).

Specifically, Claim 1 recites, among other elements:

**receiving, at a merge utility** executing on a computer system, **a request to merge a first merge document in a merge format with a second document in an original format;**

...

**in response to the request, the merge utility causing the second document to be converted** from the original format to the merge format to create a second merge document;  
wherein the second merge document is in the merge format;

...

the merge utility merging the first merge document and the second merge document to generate a composite merge document;  
and  
after generating the composite merge document, the merge utility causing said composite merge document to be delivered to an output device;

...

wherein the original format is a format that is not supported by the output device, and therefore needs to be converted to

another format that is supported by the output device in order to be properly interpreted by the output device; and wherein the merge format is a format that is supported by the output device, and therefore does not need to be converted to another format that is supported by the output device in order to be properly interpreted by the output device . . . .

The cited references fail to teach or suggest such a method for at least the following reasons:

- (1) Neither reference teaches that a merge utility responds to a request to merge a document in an original format with a document in a merge format.

Claim 1 recites “receiving, at a merge utility executing on a computer system, a request to merge a first merge document in a merge format with a second document in an original format.” Furthermore, one or more of the steps of Claim 1 are performed “in response to the request.” Neither *Barry*’s nor *Scwhier*’s alleged merge utilities receive or respond to such a request.

Rather, in *Barry*, the alleged merge utility responds to a request to merge two documents that are already in a merge format. *E.g. Barry* at FIG. 8 (showing both inputs to be PDL). Moreover, in *Schwier*, the only requests to merge documents appear to be requests to merge variable and static data in an original format at application 10 and to merge variable and static data in a merge format at printer 7. *E.g. Schwier* at FIG. 2; col. 5, lines 30–38; col. 7, lines 4–9. In neither instance is only one of the two input documents in a merge format.

Thus, neither *Barry* nor *Schwier* teach “receiving [and responding to], at a merge utility executing on a computer system, a request to merge a first merge document in a merge format with a second document in an original format.

- (2) Neither reference features a merge utility that causes a document to be converted from an original format to a merge format

Claim 1 further recites that “in response to the request, the merge utility caus[es] the second document to be converted from the original format to the merge format to create a second merge document.” Neither *Barry*’s nor *Scwhier*’s alleged merge utilities cause an input document to be converted from an original format to a merge format.

The Office Action already acknowledges that *Barry* is silent as to a conversion of a document from an original format to a merge format, within the meaning of Claim 1. The Office Action instead points out that the conversion of documents from an original format to a merge format is well known, as illustrated in *Schwier* at FIG. 8. Even so, *Schwier* does not teach that **“the merge utility caus[es] the second document to be converted** from the original format to the merge format.” In *Schwier*, conversion from an original format to a merge format is caused by a PCL converter 18. *Schwier* at FIG. 2; col. 7, 7–9; *see also* FIG. 9 (illustrating the converter as “EMF->PCL Convertor 58”). This converter is not a merge utility. In fact, it functions in the opposite manner of a merge utility, in that it filters or splits a single data stream into multiple data streams. *Schwier* at FIG. 2; col. 7, 7–9. Meanwhile, the only components of *Schwier* that do perform a merger, application 10 and printer 7, do not cause or perform any conversion operations.

For at least the foregoing reasons, the combination of *Barry* and *Schwier* fails to provide the complete subject matter recited in independent Claim 1. Therefore, the combination of *Barry* and *Schwier* would not have rendered Claim 1 obvious under 35 U.S.C. § 103. Reconsideration is respectfully requested.

## CLAIM 9

Claim 9 recites the method of Claim 1, wherein:

causing the second document to be converted from the original format to the merge format to create the second merge document includes:  
the merge utility generating, based on the original format, a set of conversion instructions to convert the second document into said second merge document;  
**passing the set of conversion instructions from the merge utility to the first document authoring application;**  
wherein the conversion instructions, when interpreted by the first document authoring application, **cause the first document authoring application to generate the second merge document** based on said set of conversion instructions.

Thus, Claim 9 teaches that, in response to the merge request recited in Claim 1, a merge utility passes conversion instructions to the document authoring application in which the second

document was created. These instructions cause the document authoring application to convert the second document from the original format to the merge format.

The cited references fail to teach or suggest such a method, for at least the reasons that, as explained above, the references fail to teach the merge request of Claim 1.

The Office Action alleges that certain steps of Claim 9 are taught in *Schwier* at FIG. 2 and column 4, lines 15–20. The Office Action is mistaken, for at least the reason that, while this passage of *Schwier* may state that an application sends an instruction to a PCL convertor 18 to convert a document, **PCL convertor 18 is not a “first document authoring application.”** Moreover, the application does not pass the instruction to the PCL convertor until after the application has already performed a merge operation, whereas Claim 9 recites that this step occurs as part of converting the second document prior to merging the first merge document and the second merge document. In other words, the instructions in *Schwier* are to convert an already combined document, not a single input document as recited in Claim 9.

For at least the foregoing reasons, the combination of *Barry* and *Schwier* fails to provide the complete subject matter recited in independent Claim 9. Therefore, the combination of *Barry* and *Schwier* would not have rendered Claim 9 obvious under 35 U.S.C. § 103. Reconsideration is respectfully requested.

### CLAIM 10

The method of Claim 10 recites the method of Claim 1, wherein, among other elements:

the request contains information about the first document authoring application

The method of Claim 10 further recites, among other elements:

the merge utility generating, based on the information about the first document authoring application, a set of conversion instructions to convert the second document into said second merge document;  
passing the set of conversion instructions from the merge utility to the first document authoring application; and  
wherein the conversion instructions, when interpreted by the first document authoring application, cause the first document

authoring application to generate the second merge  
document based on said set of conversion instructions

Therefore, the method of Claim 10 recites that a merge utility generates conversion instructions based at least upon information about a document authoring application that was included in the original merge request. The conversion instructions are used to cause the conversion of the second document by the document authoring application, similar to Claim 9.

The cited references fail to teach or suggest such a method, for at least the reasons that, as explained above, the references fail to teach the merge request of Claim 1. The Office Action nonetheless alleges that *Schwier*'s merge request "contains information about the first document authoring application" because *Schwier* states in col. 4, lines 25–26, that "the referencing is thereby particularly controlled via data that are input via a user interface." Clearly, the Office Action is in error. This passage has nothing to do with a merge request, much less "information about the first document authoring application." Clearly, the passage cannot teach or suggest that **a merge request "contains information about the first document authoring application."**

The Office Action further alleges that *Schwier* teaches that a set of conversion instructions is generated "based on the information about the document authoring application" included in the merge request because of "the program code or device which enables the PCL converter 18 in Figure 2." Again, the Office Action is clearly in error. *Schwier* does not teach or suggest that the **"program code or device which enables the PCL converter 18" is generated "based on the information about the document authoring application" included in the merge request.**

The Office Action further alleges that *Schwier* teaches the latter steps of this method in col. 9, lines 59–62 and 65–67. The Office Action is again mistaken. The relied upon passage of *Schwier* states "Print processor 49a does not forward the EMF data directly to port monitor 51 but calls the converter unit 58, wherein the EMF data stream is converted into a PCL print data stream." In other words, **this passage states that a print processor—which is not a merge utility—sends instructions to a convertor unit—which is not a document authoring application—to perform a conversion.** This passage therefore teaches absolutely nothing about "passing the set of conversion instructions from the merge utility to the first document authoring application," as recited in Claim 9. Moreover, this passage does not teach that the

conversion instructions “cause the first document authoring application to generate the second merge document.”

For at least the foregoing reasons, the combination of *Barry* and *Schwier* fails to provide the complete subject matter recited in independent Claim 9. Therefore, the combination of *Barry* and *Schwier* would not have rendered Claim 9 obvious under 35 U.S.C. § 103. Reconsideration is respectfully requested.

### **CLAIM 15**

Independent Claim 15 also recites features argued above with relation to Claim 1, although Claim 15 is expressed in another format. Because Claim 15 has at least one of the features described above for Claim 1, Claim 15 is therefore allowable over the combination of *Barry* and *Schwier* for at least one of the same reasons as given above for Claim 1. Reconsideration is respectfully requested.

### **DEPENDENT CLAIMS 2–8, 11–14, AND 16–32**

Each of 2–8, 11–14, and 16–32 depends from Claim 1 or 15, and includes the above-quoted features of its parent claim by dependency. Thus, the combination of *Barry* and *Schwier* also fails to teach or suggest at least one feature found in Claims 2–8, 11–14, and 16–32. Therefore, the combination of *Barry* and *Schwier* does not render obvious Claims 2–8, 11–14, and 16–32. Reconsideration of the rejection is respectfully requested.

In addition, each of Claims 2–8, 11–14, and 16–32 recites at least one feature that independently renders it patentable. However, to expedite prosecution in light of the fundamental differences already identified, further arguments for each independently patentable feature of Claims 2–8, 11–14, and 16–32 are not provided at this time. Applicants reserve the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

## II. CONCLUSION

For the reasons set forth above, all of the pending claims are now in condition for allowance. The Examiner is respectfully requested to contact the undersigned by telephone relating to any issue that would advance examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: January 11, 2010

/KarlTRees#58983/

Karl T. Rees, Reg. No. 58,983

2055 Gateway Place, Suite 550  
San Jose, CA 95110  
Telephone: (408) 414-1233  
Facsimile: (408) 414-1076